

# JAVA.UTIL.OBSERVABLE CLASS

[http://www.tutorialspoint.com/java/util/java\\_util\\_observable.htm](http://www.tutorialspoint.com/java/util/java_util_observable.htm)

Copyright © tutorialspoint.com

## Introduction

The **java.util.Observable** class represents an observable object, or "data" in the model-view paradigm. Following are the important points about Observable:

- The class can be subclassed to represent an object that the application wants to have observed.
- An observable object can have one or more observers.

## Class declaration

Following is the declaration for **java.util.Observable** class:

```
public class Observable
    extends Object
```

## Class constructors

| S.N. | Constructor & Description   |
|------|---|
| 1    | <b>Observable()</b><br>This constructs an Observable with zero Observers. |

## Class methods

| S.N. | Method & Description   |
|------|--|
| 1    | <a href="#"><u>void addObserver(Observer o)</u></a><br>This method adds an observer to the set of observers for this object, provided that it is not the same as some observer already in the set.   |
| 2    | <a href="#"><u>protected void clearChanged()</u></a><br>This method indicates that this object has no longer changed, or that it has already notified all of its observers of its most recent change, so that the hasChanged method will now return false. |
| 3    | <a href="#"><u>int countObservers()</u></a><br>This method returns the number of observers of this Observable object.  |
| 4    | <a href="#"><u>void deleteObserver(Observer o)</u></a><br>This method deletes an observer from the set of observers of this object.  |
| 5    | <a href="#"><u>void deleteObservers()</u></a><br>This method clears the observer list so that this object no longer has any observers.   |
| 6    | <a href="#"><u>boolean hasChanged()</u></a><br>This method tests if this object has changed.   |

|   |  |
|---|--|
| 7 | <a href="#"><u>void notifyObservers()</u></a><br>If this object has changed, as indicated by the hasChanged method, then notify all of its observers and then call the clearChanged method to indicate that this object has no longer changed.           |
| 8 | <a href="#"><u>void notifyObservers(Object arg)</u></a><br>If this object has changed, as indicated by the hasChanged method, then notify all of its observers and then call the clearChanged method to indicate that this object has no longer changed. |
| 9 | <a href="#"><u>protected void setChanged()</u></a><br>This method returns marks this Observable object as having been changed; the hasChanged method will now return true.   |

## Methods inherited

This class inherits methods from the following classes:

- java.util.Object